INFORMATION AND CONTROL 23, 48–57 (1973)

Some Remarks on State Grammars and Matrix Grammars

ETSURO MORIYA

Department of Computer Science, The University of Electro-communications, Chofu-shi, Tokyo, Japan

The context-free matrix grammar and the state grammar without any restriction in applying productions are considered. It turned out that these grammars are equivalent in the generative power. Another type of state grammar called the state grammar with unconditional transfer is introduced, and it is shown that each context-free matrix language is a homomorphic image of the intersection of a state language with unconditional transfer and a regular set.

INTRODUCTION

Recently various ways of restrictions on Chomsky generative grammars were introduced and studied, such as matrix grammars (Abraham, 1965), grammars with control set (language) (Ginsburg and Spanier, 1968), grammars with partial ordering of the rules (Friš, 1968), programmed grammars (Rosenkrantz, 1969), scattered context grammars (Greibach and Hopcroft, 1969), periodically time-variant grammars (Salomaa, 1970), random context grammars (Van der Walt, 1970), state grammars (Kasai, 1970), and unordered scattered context grammars (Milgram and Rosenfeld, 1971; Mayer 1972). They all fall into a same category in the sense that the restricted use of productions is an essential requirement.

Salomaa (1970) showed that matrix grammars, programmed grammars, grammars with regular control language, and periodically time-variant grammars are all equivalent with respect to the generative power under the same condition in applying a production, and Milgram and Rosenfeld (1971) and Mayer (1972) showed that unordered scattered context grammars are also equivalent to matrix grammars.

In this paper we shall treat the state grammar which is operating under (what we call) the free interpretation, and show that this restricted state grammar is also equivalent in the generative power to these equivalent grammars.[1] Next, another type of state grammar called the state grammar with unconditional transfer will be introduced, and the relationship between state languages under the free interpretation (equivalently context-free matrix languages) and state languages with unconditional transfer will be given in terms of homomorphism and intersection with a regular set.

The author would like to thank Professor Arto Salomaa, University of Turku, who found out a serious error in the manuscript of this paper.

1. Definitions and Equivalence

We briefly state the basic definitions and notations to be used in this paper. For a shortness, we employ, for the most part, the conventional terms and symbolisms from Ginsburg (1966). Also for the definitions concerning the *state grammar* and the *programmed grammar*, the reader is referred to Kasai (1970) and Rosenkrantz (1969), respectively.

First we define a special state grammar.

Definition. Let $G = (K, V, \Sigma, P, p_0, S)$ be a state grammar in which productions of the form $(p, X) \to (q, \epsilon)$ are allowed.[2,3] A *configuration* of G is an element of $K \times V^*$. For a state production $\pi : (p, X) \to (q, u)$, two configurations (p, w_1) and (q, w_2) are *related with respect to* π if there exist x and y in V^* such that $w_1 = xXy$ and $w_2 = xuy$. In this case we write $(p, w_1) \Rightarrow_\pi (q, w_2)$.[4] When π is understood, it may be omitted from $\Rightarrow_\pi$. $\overset{*}{\Rightarrow}$ is the transitive closure of $\Rightarrow$, that is, $(s, u) \overset{*}{\Rightarrow} (t, v)$ if and only if there exist $s_0, \ldots, s_k$ in K and $u_0, \ldots, u_k$ in V^* for some $k \geqslant 1$ such that $s_0 = s$, $s_k = t$, $u_0 = u$, $u_k = v$, and (s_i, u_i) is related to (s_{i+1}, u_{i+1}) with respect to some π_{i+1} in P for each $0 \leqslant i < k$. A sequence $(s_0, u_0) \Rightarrow \cdots \Rightarrow (s_k, u_k)$ of related configurations is a *derivation* in G. A state grammar in which derivations are so defined is said to be *operating under the free interpretation*.

[1] The proof of some of these equivalences was obtained in 1970. After that, we found the paper establishing the same result (Salomaa, 1970). So that, in this paper we show only the equivalence of the state grammar operating under the free interpretation and the context-free matrix grammar.

[2] In the original definition of the state grammar, productions of the form $(p, X) \to (q, \epsilon)$ are not allowed.

[3] ϵ is the empty word.

[4] Recall that in a state grammar operating not under the free interpretation (i.e., in Kasai's original definition), X must be the left-most applicable variable under p.

The language generated by a state grammar G which is operating under the free interpretation (called a *fisl*) is defined to be

$$L_f(G) = \{w \text{ in } \Sigma^* \mid (p_0, S) \stackrel{*}{\Rightarrow} (q, w), q \text{ in } K\}.$$

DEFINITION. A state grammar is *ϵ-free* if no production is of the form $(p, X) \to (q, \epsilon)$.

DEFINITION. A state grammar *with accepting states* (abbreviated *a-state grammar*) is a 7-tuple $G = (K, V, \Sigma, P, p_0, S, F)$, where

$$G' = (K, V, \Sigma, P, p_0, S)$$

is a state grammar and F is a subset of K. The language generated by G under the free interpretation is the set

$$L_f(G) = \{w \text{ in } \Sigma^* \mid (p_0, S) \stackrel{*}{\Rightarrow} (q, w), q \text{ in } F\}.$$

LEMMA 1.1. *$L = L_f(G)$ for some (ϵ-free) a-state grammar G if and only if $L = L_f(G')$ for some (ϵ-free) state grammar G'.*

We now define the matrix grammar in our notation. Our matrix grammar is equivalent to the context-free matrix grammar of Salomaa (1970).

DEFINITION. A (context-free)[5] *matrix grammar* is a pair (G, M), where $G = (V, \Sigma, P, S)$ is a context-free grammar and M is a finite subset of P^+.[6] Each element of M is called a *matrix production*. (G, M) is *ϵ-free* if G is so.

Let w and w' be any words in V^* and π a production $X \to u$ in P. If there are x and y in V^* such that $w = xXy$ and $w' = xuy$, we say $w \Rightarrow_\pi w'$ (or simply $w \Rightarrow w'$). If $\beta = \pi_1 \cdots \pi_k$ with each π_i in P and if there exist $w_1, \ldots, w_{k-1}$ in V^* such that $w_0 \Rightarrow_{\pi_1} w_1 \Rightarrow_{\pi_2} \cdots \Rightarrow_{\pi_{k-1}} w_{k-1} \Rightarrow_{\pi_k} w_k$, then we say $w_0 \stackrel{*}{\Rightarrow}_\beta w_k$ (or $w_0 \stackrel{*}{\Rightarrow} w_k$). The language generated by a matrix grammar (G, M), called a *matrix language*, is defined to be

$$L_f(G, M) = \{w \text{ in } \Sigma^* \mid S \underset{\alpha}{\stackrel{*}{\Rightarrow}} w, \alpha \text{ in } M^+\}.$$

Now the equivalence of fisl's and matrix languages will be briefly stated. We begin with showing that the class of languages generated by ϵ-free matrix grammars is closed under restricted homomorphism.

DEFINITION. A homomorphism h from $\Sigma_1{}^*$ into $\Sigma_2{}^*$ is *k-restricted* on a subset L of $\Sigma_1{}^*$ if $h(w) = \epsilon$ implies $w = \epsilon$ and for each w in L, $h(w') \neq \epsilon$

[5] The specification "context-free" will be disregarded in this paper.

[6] $V^+ = V^* - \{\epsilon\}$.

for each subword w' of w such that the length of w' is greater than or equal to k. A class of languages $\mathscr{L}$ is *closed under restricted homomorphism* if $h(L)$ is in $\mathscr{L}$ whenever $L \subseteq \Sigma_1{}^*$ is in $\mathscr{L}$ and h is a homomorphism of $\Sigma_1{}^*$ which is $k - 1$ restricted on L for some k.

DEFINITION. A matrix grammar (G, M) is in *binary form* if all productions of G are of the forms $X \to YZ$, $X \to Y$, $X \to a$, or $X \to \epsilon$, where X, Y, Z are variables and a is a terminal symbol.

It is easy to show that for each (ϵ-free) matrix grammar (G, M), there exists a (ϵ-free) matrix grammar (G', M') which is in binary form such that $L_f(G, M) = L_f(G', M')$.

LEMMA 1.2. *If $L = L_f(G, M) \subseteq \Sigma^+$ for some ϵ-free matrix grammar (G, M) and if h is a homomorphism of Σ^* which is k-restricted on L, then $h(L) = L_f(G', M')$ for some ϵ-free matrix grammar (G', M').*

Proof. Let $G = (V, \Sigma, P, S)$. We may assume that (G, M) is in binary form. Consider the grammar $G' = (V', \Sigma', P', S')$, where $V' = \{[\alpha] \mid \alpha$ in V^*, $|\alpha| \leqslant 2k\} \cup \{S'\}$,[7] Σ' consists of those symbols found in words w such that $h(a) = w$ for some a in Σ, and P' is defined as follows. Let $N_1 = \{[\alpha] \mid \alpha$ in V^*, $k \leqslant |\alpha| \leqslant 2k - 1\}$ and $N_2 = \{[\alpha] \mid \alpha$ in V^*, $k \leqslant |\alpha| \leqslant 2k\}$. Then

(1) $S' \to h(\alpha)$ is in P' for each α in $L \cap \{\beta$ in $\Sigma^+ \| \beta | < k\}$.

(2) $S' \to [\alpha]$ is in P' for each α in $\{\beta$ in $V^* \mid S \overset{*}{\Rightarrow}_\tau \beta$, $k \leqslant |\beta| \leqslant 2k$, τ in $M^+\}$.

(3) $[\alpha X \beta] \to [\alpha u \beta]$ is in P' for each $[\alpha X \beta]$ in N_1 if $X \to u$ is in P.

(4) $[\alpha] \to [\beta][\gamma]$ is in P' if α is in N_2, and β and γ are in N_1 with $\alpha = \beta\gamma$.

(5) $[\alpha] \to h(\alpha)$ is in P' for each α in $N_1 \cap \{[\alpha] \mid \alpha$ in $\Sigma^+\}$.

Now let M' consist of:

(6) Each production in (1), (2), and (5).

(7) $\eta_1\theta_1 \cdots \eta_n\theta_n$ if $\pi_1 \cdots \pi_n$ is in M, where η_i is one of those $[\alpha X \beta] \to [\alpha u \beta]$ in (3) constructed from $\pi_i : X \to u$, and θ_i is empty or one of those in (4).

Clearly (G', M') is an ϵ-free matrix grammar and $L_f(G', M') = h(L)$.

THEOREM 1.1. *$L = L_f(G_1)$ for some (ϵ-free) state grammar G_1 if and only if $L = L_f(G_2, M_2)$ for some (ϵ-free) matrix grammar (G_2, M_2).*

[7] $|w|$ denotes the length of a word w.

Proof. Given a (ϵ-free) state grammar G_1, it is easy to construct a (ϵ-free) matrix grammar which generates $(L_f(G_1)c)L_f(G_1)$ (with c a new symbol). In the former case, use Lemma 1.2.

Conversely, given a (ϵ-free) matrix grammar (G_2, M_2), an (ϵ-free) a-state grammar G_3 is easily found such that $L_f(G_3) = L_f(G_2, M_2)$. Then use Lemma 1.1.

Remark. Although not explicitly stated, all constructions in this paper are effective.

2. State Grammars with Unconditional Transfer

Now we consider another type of state grammars.

Definition. Let $G = (K, V, \Sigma, P, p_0, S)$ be a state grammar which is operating under the free interpretation. G is a state grammar *with unconditional transfer* (abbreviated *utsg*) if the relation $\Rightarrow$ is defined as follows. Let (p, w_1) and (q, w_2) be configurations of G, and let π be a state production $(p, X) \to (q, u)$. We write[8] $(p, w_1) \Rightarrow_\pi (q, w_2)$ if either

(1) w_1 contains no X and $w_2 = w_1$, or

(2) there exist x and y in V^* such that $w_1 = xXy$ and $w_2 = xuy$. $\overset{*}{\Rightarrow}$ is the transitive closure of $\Rightarrow$.

The language generated by a utsg G (called a *utsl*) is denoted by $L_u(G)$, that is,

$$L_u(G) = \{w \text{ in } \Sigma^* \mid (p_0, S) \overset{*}{\Rightarrow} (q, w), q \text{ in } K\}.$$

Theorem 2.1. *The class of languages generated by ϵ-free* utsg's *is identical to the class of languages generated by* utcfpg's *which are operating under the free interpretation.*[9]

Proof. It can be easily shown that, for each ϵ-free utsg G, there exists an ϵ-free utsg G' with the following properties:

(a) $L_u(G) = L_u(G')$.

(b) If $(p, X) \to (q, u)$ and $(p, Y) \to (r, v)$ are state productions of G', then $X = Y$ and $u = v$.

[8] Thus each utsg is operating under the free interpretation.

[9] See Rosenkrantz (1969).

Thus each language generated by an ϵ-free utsg can be generated by a utcfpg operating under the free interpretation. The converse inclusion is obvious.

COROLLARY 2.1. *The emptiness problem for (ϵ-free)* utsg'*s is solvable.*

Proof. Rosenkrantz (1969) showed that the emptiness problem for utcfpg's is solvable. The proof works also for utcfpg's operating under the free interpretation. Note that the emptiness problems for ϵ-free utsg's and for (not necessarily ϵ-free) utsg's are equivalent.

Since the class of languages generated by (ϵ-free) utsg's is closed under union, the next corollary follows from the fact (Bar-Hillel *et al.*, 1961) that for (ϵ-free) context-free grammars G_1 and G_2, it is undecidable whether or not $L(G_1) \cap L(G_2)$ is empty.

COROLLARY 2.2. *The class of languages generated by (ϵ-free)* utsg'*s is not effectively closed under both intersection and complementation.*

We now establish a relationship between utsl's and fisl's (equivalently matrix languages, by Theorem 1.1) in terms of homomorphism and intersection with a regular set.

THEOREM 2.2. *For each (ϵ-free) state grammar G, there exist a (ϵ-free)* utsg *G', a regular set R, and a homomorphism h such that $L_f(G) = h(L_u(G') \cap R)$.*

Proof.[10] Let $G = (K, V, \Sigma, P, p_0, S)$ and $L = L_f(G)$. Suppose that $V - \Sigma = \{X_1, ..., X_n\}$, $n \geqslant 1$. Let $p_0', f, S', \#, c$, and d be new symbols and let $K' = K \cup \{p_0', f\}$, $V' = V \cup \{S', \#, c, d\}$, $\Sigma' = \Sigma \cup \{\#, c, d\}$. Consider the grammar $G' = (K', V', \Sigma', P', p_0', S')$, where P' consists of

(1) $(p_0', S') \to (p_0, S \# X_1 \cdots X_n)$;

(2) $(p, X) \to (q, cu)$ if $(p, X) \to (q, u)$ is in P;

(3) $(p, X_1) \to (f, d)$ for all p in K;

(4) $(f, X_i) \to (f, d)$ for all i, $2 \leqslant i \leqslant n$;

Let $L' = L_u(G')$ and $R = (\Sigma \cup \{c\})^+ \# d^n$. Finally let h be the homomorphism of $(V')^*$ defined by $h(x) = x$ for each x in $V' - \{\#, c, d\}$ and $h(\#) = h(c) = h(d) = \epsilon$. We shall show that $L = h(L' \cap R)$.

[10] In this proof, the introduction of the symbol c is essential.

Let w be any word in L. There exists a derivation in G generating w:

$$(p_0, S) \underset{\pi_1}{\Rightarrow} (p_1, w_1) \underset{\pi_2}{\Rightarrow} \cdots \underset{\pi_r}{\Rightarrow} (p_r, w_r) = (p, w).$$

For each i, $1 \leqslant i \leqslant r$, let ξ_i be $(p, X) \to (q, cu)$ if π_i is $(p, X) \to (q, u)$. Clearly

$$\begin{aligned}(p_0', S') &\Rightarrow (p_0, S \# X_1 \cdots X_n) \underset{\xi_1}{\Rightarrow} (p_1, y_1 \# X_1 \cdots X_n) \underset{\xi_2}{\Rightarrow} \cdots \\ &\underset{\xi_r}{\Rightarrow} (p_r, y_r \# X_1 \cdots X_n) \Rightarrow (f, y_r \# dX_2 \cdots X_n) \overset{*}{\Rightarrow} (f, y_r \# d^n)\end{aligned}$$

is valid in G', where $h(y_i) = w_i$ for each i, $1 \leqslant i \leqslant r$. Thus w is in $h(L' \cap R)$.

Conversely let w be in $h(L' \cap R)$. There exists w' in $L' \cap R$ such that $w = h(w')$. Since w' is in L', there is a derivation

$$(p_0', S') \underset{\tau_1}{\Rightarrow} (q_1, w_1) \underset{\tau_2}{\Rightarrow} \cdots \underset{\tau_s}{\Rightarrow} (q_s, w_s) = (q, w') \qquad (*)$$

in G'. Clearly $q_1 = p_0$ and $w_1 = S \# X_1 \cdots X_n$. Let k be the smallest integer such that $q_k = f$. Then each τ_j, $1 < j < k$, must be of the form (2), and τ_k must be of the form (3). We can write $w_k = y \# x$ for some y and x. From the definition of P', the state once entered f never change, and the state production whose left-side state is f can only replace a variable by d. Thus in view of w' being in $L' \cap R$, y is in $(\Sigma \cup \{c\})^+$ and $x = dX_2 \cdots X_n$. Furthermore the latter half part of (*) should be

$$(q_k, w_k) = (f, y \# dX_2 \cdots X_n) \underset{\tau_{k+1}}{\Rightarrow} \cdots \underset{\tau_s}{\Rightarrow} (f, y \# d^n) = (q, w').$$

Now $x = dX_2 \cdots X_n$ implies that each w_i, $1 \leqslant i < k$, contains a string $X_1 \cdots X_n$ on the right of $\#$, that is, $w_i = y_i \# X_1 \cdots X_n$ for some y_i, which means that each τ_i is actually applied to w_{i-1}; in other words, it is not used only to change the state. Thus

$$(p_0, S) = (q_1, z_1) \underset{\eta_2}{\Rightarrow} \cdots \underset{\eta_{k-1}}{\Rightarrow} (q_{k-1}, z_{k-1})$$

is valid in G with $z_{k-1} = h(y)$, where η_i is $(p, X) \to (q, u)$ if τ_i is $(p, X) \to (q, cu)$. Hence $w = h(w') = h(y \# d^n)$ is in L.

By this theorem, it follows that if the class of utsl's is closed under intersection with a regular set, then the emptiness problem for fisl's (matrix languages) is solvable.

THEOREM 2.3. *The class of languages generated by* (ϵ-*free*) utsg's *is closed under* (ϵ-*free*) *substitution.*

Proof. Let $L = L_u(G)$, where $G = (K, V, \Sigma, P, p_0, S)$ is a (ϵ-free) utsg, and let τ be a (ϵ-free) substitution. For each a in Σ, let $\tau(a) = L_u(G_a)$, where $G_a = (K_a, V_a, \Sigma_a, P_a, p_a, S_a)$ is a (ϵ-free) utsg. There is no loss of generality in assuming that

$$(V_a - \Sigma_a) \cap (V_b - \Sigma_b) = \phi, \quad (V_a - \Sigma_a) \cap (V - \Sigma) = \phi, \quad K_a \cap K_b = \phi,$$

and $K_a \cap K = \phi$ for all $a \neq b$ in Σ. Assume $V_a - \Sigma_a = \{X_{a,1}, \ldots, X_{a,n(a)}\}$. For each a in Σ, let $\bar{a}$ and $f_{a,i}$, $1 \leqslant i \leqslant n(a)$, be new symbols. For further new symbols f and D, let $K' = K \cup \bigcup_{a \text{ in } \Sigma} K_a \cup \{f_{a,i} \mid a \text{ in } \Sigma, 1 \leqslant i \leqslant n(a)\} \cup \{f\}$, $V' = V \cup \bigcup_{a \text{in} \Sigma} V_a \cup \{D\}$ and $\Sigma' = \bigcup_{a \text{ in } \Sigma} \Sigma_a$. Let h be the homomorphism of V^* defined by $h(X) = X$ for each X in $V - \Sigma$ and $h(a) = \bar{a}$ for each a in Σ. Finally let $G' = (K', V', \Sigma', P', p_0, S)$ be a (ϵ-free) utsg, where P' contains:

(1) $(p, X) \to (q, h(u))$ for all $(p, X) \to (q, u)$ in P.

(2) $(p, \bar{a}) \to (p_a, S_a)$ for all p in K and all a in Σ.

(3) All productions in $\bigcup_{a \text{ in } \Sigma} P_a$.

(4) For all a and b in Σ,

- (a) $(q, D) \to (f_{a,1}, D)$ for all q in K_a;
- (b) $(f_{a,i}, X_{a,i}) \to (f_{a,i+1}, D)$ for all i, $1 \leqslant i < n(a)$;
- (c) $(f_{a,n(a)}, X_{a,n(a)}) \to (f, D)$;
- (d) $(f, \bar{b}) \to (p_b, S_b)$.

Note that the requirement of productions (4) is essential. Since the state once entered $K' - K$ never go back again to K, they guarantee that derivations in G, G_a, and G_b are independently realized of each other, serving as subderivations of a derivation in G'. Thus it is a straightforward matter to show that $L_u(G') = \tau(L)$.

COROLLARY 2.3. *The class of languages generated by (ϵ-free)* utsg's *is closed under union, concatenation, Kleene (+) *, and (ϵ-free) homomorphism.*

3. Some Additional Properties of fisl's

In this section some additional results will be summarized. Most of them are by a standard technique, and some are stated in Salomaa (1969, 1970).

THEOREM 3.1. *The class of* fisl's *(matrix languages)*

(a) *is closed under union, concatenation, word reversal, intersection with a regular set, substitution by context-free languages and permutations, and*

(b) *is closed under* gsm *mappings, inverse* gsm *mappings, and inverse homomorphism.*

Proof. (a) Only the closure under permutations will be given. For a state grammar $G = (K, V, \Sigma, P, p_0, S)$, let

$$\bar{\Sigma} = \{\bar{a} \mid a \text{ in } \Sigma\}, \quad K' = K \cup K \times \Sigma \times \Sigma,$$

and $V' = V \cup \bar{\Sigma}$. Let h be the homomorphism of V^* into $(V')^*$ which maps each X in $V - \Sigma$ into itself and each a in Σ into $\bar{a}$. Consider the grammar $G' = (K', V', \Sigma, P', p_0, S)$ with the productions P':

(1) $(p, X) \to (q, h(u))$ for all $(p, X) \to (q, u)$ in P.

(2) $(p, \bar{a}) \to ([p, a, b], \bar{b})$ for all p in K and a, b in Σ.

(3) $([p, a, b], \bar{b}) \to (p, \bar{a})$ for all p in K and a, b in Σ.

(4) $(p, \bar{a}) \to (p, a)$ for all p in K and a in Σ.

Clearly $L_f(G')$ contains all and only those words which are permutations of words in $L_f(G)$.

(b) Every class of languages closed under substitution by finite sets, and intersection with a regular set is closed under gsm mappings (Hopcroft and Ullman, 1969). Every class of languages closed under restricted homomorphism, substitution by regular sets, and union and intersection with a regular set, is closed under inverse gsm mappings (Hopcroft and Ullman, 1969). Inverse homomorphism is a special case of inverse gsm mappings. Hence the assertion follows from (a) and Lemma 1.2.

Remark. The above theorem also holds for the class of languages generated by ϵ-free state grammars operating under the free interpretation, if the corresponding operations are ϵ-free.

ACKNOWLEDGMENT

The author wishes to express his gratitude to Professor Hiroshi Noguchi of Waseda University. He is also grateful to Professor Takumi Kasai of Kyoto University for his helpful discussion with the author.

References

ABRAHAM, S. (1965), Some questions of phrase structure grammars I, *Comput. Linguistics* **4**, 61–70.

BAR-HILLEL, Y., PERLES, M., AND SHAMIR, E. (1961), On formal properties of simple phrase structure grammars, *Z. Phonetik, Sprachwiss. Kommunikat.* **14**, 143–172.

FRIŠ, I. (1968), Grammars with partial ordering of the rules, *Inform. Control* **12**, 415–425.

GINSBURG, S. (1966), "The Mathematical Theory of Context-Free Languages," McGraw-Hill, New York.

GINSBURG, S. AND SPANIER, E. H. (1968), Control sets on grammars, *Math. Systems Theory* **2**, 159–177.

GREIBACH, S. AND HOPCROFT, J. (1969), Scattered context grammars, *J. Comput. System Sci.* **3**, 233–247.

HOPCROFT, J. E. AND ULLMAN, J. D. (1969), "Formal Languages and Their Relation to Automata," Addison-Wesley, Reading, Mass.

KASAI, T. (1970), An hierarchy between context-free and context-sensitive languages, *J. Comput. System Sci.* **4**, 492–508.

MAYER, O. (1972), Some restrictive devices for context-free grammars, *Inform. Control* **20**, 69–92.

MILGRAM, D. L. AND ROSENFELD, A. (1971), A note on scattered context grammars, *Inform. Processing Letters* **1**, 47–50.

SALOMAA, A. (1969), On grammars with restricted use of productions, *Ann. Acad. Sci. Fenn., Series* A **1**, 454.

SALOMAA, A. (1970), Periodically time-variant context-free grammars, *Inform. Control* **17**, 294–311.

ROSENKRANTZ, D. J. (1969), Programmed grammars and classes of formal languages, *J. Assoc. Comput. Mach.* **16**, 107–131.

VAN DER WALT, A. P. J. (1970), Random context languages, *Symposium on formal languages*, 1970, in Oberwolfach, Germany.